Lab Manual: 09

Lab Topic: Abstract Class and Interface

Course Code: CSE1116 (Object Oriented Programming Laboratory)

Course Instructor: Mir Moynuddin Ahmed Shibly, Lecturer, CSE

Lab Objective

1. **Write** the definition of the super class and extend it to create multiple subclasses.
2. **Write** codes to implement polymorphism.

Lab Activities

A. Abstract Class

- Create an abstract class 'Shape' with one attribute named ‘area’ of double type.
- Write proper setter and getter for the attributes
- Three abstract methods namely 'RectangleArea' taking two parameters, 'SquareArea' and 'CircleArea' taking one parameter each.
- The parameters of 'RectangleArea' are its length and breadth, that of 'SquareArea' is its side and that of 'CircleArea' is its radius.
- Now create another class 'Area' containing all the three methods 'RectangleArea', 'SquareArea' and 'CircleArea' for calculating the area of rectangle, square and circle respectively.
- Create an ArrayList of Shape type. Your main () method must display the following first:

```
Press (1) for calculating Rectangle Area
Press (2) for calculating Square Area
Press (3) for calculating Circle Area
```

- You must create at least 3 shape type reference variable and assign area type object to them in this manner
- Call the respective method for all three objects and display the area.

B. Comparable Interface

- Comparable Interface is used to compare two objects. In this problem, you'll create a class that implements the comparable interface and use it to sort an array of objects.
- Create a *Player* class with 2 fields: name of String Type and score of integer type.
- Define proper constructor to set the attributes value
- Modify the Player class to implement the Comparable interface
- Given an array of *n Player* objects and sort them in order of decreasing score; if 2 or more players have the same score, sort those players alphabetically by name. To do this, you must override the compareTo (*Player b)* method of comparable interface in the player class.

Input Format

The first line contains an integer, *n*, denoting the number of players.
Each of the *n* subsequent lines contains a player's *name* and *score*, respectively.

Output Format

Print each sorted element in the format: *namescore*

Sample Input

```
5
amy 100
david 100
heraldo 50
aakansha 75
aleksa 150
```

Sample Output

```
aleksa 150
amy 100
```

```
david 100
aakansha 75
heraldo 50
```